

42

En ce qui concerne les codes à deux lettres et autres abréviations, se référer aux "Notes explicatives relatives aux codes et abréviations" figurant au début de chaque numéro ordinaire de la Gazette du PCT.

(57) Abrégé : L'invention concerne un dispositif de reconfiguration d'un ensemble de N modules électroniques de base associés à k modules redondants, comprenant: N multiplexeurs ayant chacun une première borne (d_i) susceptible d'être connectée à l'une de $k+1$ deuxièmes bornes connectées aux $k+1$ bornes d'entrée/sortie d'un groupe ordonné de modules composé d'un module de base (U_i) et de k autres modules; $N+k$ bascules (F_i) indicatrices d'un état bon ou mauvais d'un des $N+k$ modules; et un moyen logique associé à chaque multiplexeur de rang j , j étant un entier compris entre 0 et N , pour déterminer le nombre de bascules de rang 0 à j indiquant un état mauvais, déterminer le nombre de modules du groupe ordonné associé au module de rang j , à compter pour trouver un nombre de modules bons égal au premier nombre, et connecter la première borne du multiplexeur à sa deuxième borne de rang égal au deuxième nombre.

**DISPOSITIF DE RECONFIGURATION D'UN ENSEMBLE MEMOIRE PRESENTANT
DES DEFAUTS**

La présente invention concerne un dispositif permettant de reconfigurer un ensemble de N modules électroniques associés à k modules de remplacement ou modules redondants. On considérera ici le cas où les modules électroniques sont des modules mémoire qui peuvent être des blocs d'une mémoire de grande dimension et/ou des lignes ou colonnes d'une mémoire. On gardera néanmoins à l'esprit que l'invention concerne de façon générale la reconfiguration d'un ensemble de modules électroniques.

Les figures 1A et 1B illustrent de façon très simplifiée le problème que vise à résoudre la présente invention. On considère quatre blocs de cellules mémoire M_0 à M_3 . En pratique, il s'agira par exemple de colonnes d'une mémoire pouvant comprendre plus de mille colonnes. Il est possible que certains de ces blocs mémoire présentent un défaut. Alors, on prévoit de remplacer le bloc en défaut par un bloc de remplacement. On a représenté deux blocs de remplacement S_0 et S_1 . En fait, on pourra prévoir plusieurs dizaines de blocs de remplacement.

Pour réaliser ce remplacement, au lieu de connecter les bornes d'entrée/sortie de données d_0 , d_1 , d_2 , d_3 directement aux blocs mémoire M_0 , M_1 , M_2 , M_3 , on prévoit de relier ces bornes aux blocs mémoire par l'intermédiaire de multiplexeurs

MUX₀, MUX₁, MUX₂, MUX₃, respectivement commandés par des blocs de commande C₀, C₁, C₂, C₃. Dans le cas où il y a k blocs de remplacement, chaque multiplexeur comprendra k+1 sorties. Dans l'exemple des figures 1A et 1B, on suppose que le deuxième, M₁, des blocs mémoire de base et le premier, S₀, des blocs mémoire de remplacement sont en défaut.

Dans le mode de réalisation de la figure 1A, on propose de remplacer chaque bloc mémoire défaillant par son voisin non défaillant le plus proche. On réalise une substitution "de proche en proche". Chaque multiplexeur est susceptible d'être relié au bloc du même rang et aux deux blocs de rang immédiatement supérieur. Dans le cas de la figure 1A, l'entrée d₀ est reliée au bloc M₀, l'entrée d₁ est reliée au bloc M₂ (au lieu du bloc M₁), l'entrée d₂ est reliée au bloc M₃ (au lieu du bloc M₂), et l'entrée d₃ est reliée au bloc S₁ (au lieu du bloc M₃ ou S₀). Le multiplexeur MUX₀ réalise un décalage nul, les multiplexeurs MUX₁ et MUX₂ un décalage d'un pas, et le multiplexeur MUX₃ un décalage de deux pas.

Dans le bloc de réalisation de la figure 1B, tout bloc en défaut est remplacé non pas par l'un de ses voisins non défaillant mais par l'un des blocs de remplacement. On réalise une substitution "distante". Ainsi, l'entrée d₀ est reliée au bloc M₀, l'entrée d₁ au bloc S₁ et les entrées d₂, d₃ aux blocs M₂, M₃. Ainsi, chaque multiplexeur comprend une sortie vers le bloc correspondant et deux sorties vers les blocs S₀ et S₁. Les multiplexeurs MUX₀, MUX₂ et MUX₃ réalisent un décalage nul et le multiplexeur MUX₁ un décalage de deux pas.

La présente invention vise à prévoir un dispositif de reconfiguration (de reroutage) automatique entre des bornes d'accès et des blocs mémoire non en défaut, c'est-à-dire que les blocs mémoire en défaut sont remplacés par d'autres blocs choisis selon le cas parmi les voisins du bloc défaillant (cas de la figure 1A), ou directement parmi des blocs redondants (cas de la figure 1B).

On comprendra que ce problème est relativement complexe car, pour un bloc donné, pour déterminer vers quel bloc

le reroutage doit être effectué, il faut tenir compte de l'état des blocs déjà examinés, par exemple ceux situés à droite (ou à gauche) d'un bloc considéré, et de l'état des blocs de remplacement possibles.

- 5 Pour permettre cette reconfiguration, la présente invention prévoit pour chacun des cas (décalage de proche en proche ou décalage distant) un procédé faisant appel à des opérations séquentielles ou un procédé faisant appel à un système combinatoire. De plus, dans chacun des deux cas consi-
10 dérés, la présente invention envisage que les opérations puissent être faites de manière matérielle ou logicielle.

De façon générale, la présente invention prévoit d'utiliser un algorithme selon lequel :

- on commence par le bloc situé à une première extrémité
15 d'une suite de blocs,

 on compte pour chaque bloc le nombre de blocs défaillants situés du côté de la première extrémité (les blocs défaillants comprenant éventuellement le bloc considéré lui-même),

- on associe à chaque bloc, un ensemble de blocs de
20 réserve disposés selon une suite ordonnée (les blocs voisins dans le cas de la figure 1A ou des blocs redondants dans le cas de la figure 1B),

- on compte, parmi ces blocs de réserve, des blocs successifs jusqu'à ce que l'on trouve un nombre de blocs non
25 défaillants égal au nombre de blocs défaillants précédemment comptés, et

- une fois ce résultat obtenu, on remplace le bloc considéré, par le bloc décalé de façon correspondante des blocs de réserve, les blocs non défaillants n'étant pas remplacés dans
30 le cas du passage direct aux blocs redondants.

- Ces objets, caractéristiques et avantages, ainsi que d'autres de la présente invention seront exposés en détail dans la description suivante de modes de réalisation particuliers faite à titre non-limitatif en relation avec les figures jointes
35 parmi lesquelles :

les figures 1A et 1B représentent des exemples de reconfiguration de blocs mémoire destinés à illustrer le problème posé par la présente invention ;

la figure 2 représente un exemple de réalisation de
5 dispositif de reconfiguration de blocs mémoire de type séquentiels et à décalage de proche en proche ;

la figure 3 représente une variante d'une partie du circuit logique de la figure 2 ;

la figure 4 représente un exemple de réalisation de
10 dispositifs de reconfiguration de blocs mémoire de type séquentiels et à remplacement de blocs défaillants par des blocs redondants ;

la figure 5 représente deux exemples de compteurs de 1 utilisés dans le circuit logique de la figure 4 ; et

15 les figures 6 à 10 illustrent des variantes de dispositifs de reroutage.

1. NOTATIONS

Dans la présente demande, on désignera par :

20 N : le nombre de blocs de base de la mémoire susceptible de présenter des défaillances, et

k : le nombre maximum de blocs défaillants que l'on pourra remplacer.

Un bloc de base de la mémoire initiale sera appelé U_i avec i compris entre 0 et N-1.

25 Les blocs de remplacement seront indifféremment appelés S_0 à S_{k-1} ou U_N à U_{N+k-1} .

Les multiplexeurs associés aux blocs U_0 à U_{N-1} seront appelés MUX_0 à MUX_{N-1} .

d_i désigne l'entrée d'un multiplexeur MUX_i .

30 Des bascules F_0 à F_{N-1} et S_1 à S_k sont respectivement associées aux blocs de base et de remplacement et indiquent l'état initial du bloc correspondant. Ces bascules sont à 1 si le bloc est initialement mauvais et à 0 si le bloc est initialement bon. Ces bascules sont initialisées de façon classique lors
35 d'une étape de test pendant laquelle on vérifie le fonctionnement de chaque bloc. Selon l'invention, on traite le cas où les

blocs de base aussi bien que les blocs de remplacement sont susceptibles d'être défaillants.

2. RÉPARATION DE PROCHE EN PROCHE

On considérera d'abord le cas où le processus de réparation est du type illustré en figure 1A dans lequel, à chaque multiplexeur associé à un bloc, sont également associés des blocs adjacents à ce bloc.

2.1 Reconfiguration séquentielle

La figure 2 représente un exemple de structure de reconfiguration séquentielle pour une réparation de proche en proche de cellules éventuellement défaillantes. Dans l'exemple représenté, chaque multiplexeur MUX_0 à MUX_{N-1} est associé à un compteur C_0 à C_{N-1} qui est incrémenté par un signal R_i ($R_0 \dots R_{N-1}$) produit par un circuit séquentiel de sorte que le signal R_i prend la valeur 1 pendant plusieurs cycles d'horloge consécutifs.

Les bascules F_0 à F_{N+k-1} sont initialement chargées d'un 0 si le bloc correspondant est bon et d'un 1 si le bloc correspondant est mauvais. Les signaux R_i à fournir à un circuit de commande de multiplexeur C_i dépendent de la valeur de la bascule F_i correspondante mais également des bascules de rang inférieur étant donné que, quand une position doit être décalée vers la gauche, toute position d'un rang supérieur doit également être décalée vers la gauche. La valeur de R_i dépend également des valeurs des bascules de rang supérieur à i . En effet, il faut s'assurer que le bloc que l'on vise à choisir comme bloc de remplacement est un bloc bon.

Pour résoudre ce problème, la présente invention propose l'algorithme suivant.

a) $F_i=1$ entraîne que $R_i=1$, puisque, pour un bloc U_i en défaut, il doit y avoir au moins un décalage. De plus, chaque fois qu'un signal R_i est activé, il force tous les signaux de rang supérieur à l'état actif. On doit donc avoir $R_i = F_i + R_{i-1}$.

b) Quand le signal R_i est activé ($R_i=1$), la valeur de la bascule F_{i+1} est transférée à la bascule F_i et, s'il n'y a pas de bascule F_{i+1} , un 0 est transféré à la bascule F_i . Ceci

exprime que, à la fin du présent cycle d'horloge, l'entrée d_i est connectée au bloc sur lequel l'entrée d_{i+1} était précédemment connectée. Ainsi, la bascule F_i doit indiquer l'état qui était précédemment indiqué par la bascule F_{i+1} . On déduit des points
 5 précédents l'équation séquentielle suivante pour les bascules F_i :

$F_i(t+1) = R_i.F_{i+1}(t)$, où t et $t+1$ indiquent deux instants successifs.

En résumé, l'algorithme séquentiel est :

$$R_i = R_{i-1} + F_i, \quad \text{pour } N+k-1 \geq i \geq 0 \quad (1)$$

$$F_i(t+1) = R_i.F_{i+1}(t), \quad \text{pour } N+k-1 \geq i \geq 0 \quad (2)$$

Les conditions aux limites sont que, pour la bascule R_0 , on considère que la bascule fictive R_{-1} est à l'état 0 et, pour la bascule F_{N+k-1} , on considère que la bascule fictive F_{N+k} est à l'état 0.

15 La figure 2 représente un circuit logique mettant en oeuvre les équations logiques ci-dessus.

La sortie F_i de chaque bascule est connectée à une première entrée d'une porte OU 11 dont la sortie est connectée par une porte ET 12 à l'entrée de cette bascule F_i . La sortie de
 20 la porte OU 11 fournit également le signal R_i . La deuxième entrée de la porte OU 11 est connectée à la sortie de la porte OU 11 de l'étage précédent. La sortie de la bascule F_{i+1} est renvoyée à la deuxième entrée de la porte ET 12.

Chacune des bascules est rythmée par un signal d'horloge qui commande également les compteurs C_i .
 25

On voit que ce circuit permet de satisfaire aux équations logiques susmentionnées.

On peut également comprendre le fonctionnement en considérant les étapes suivantes.

30 Au premier coup d'horloge, on suppose que les premières bascules F_0 et F_1 sont à l'état 0. Elles ne fourniront donc pas de signal de décalage aux entrées R_0 et R_1 des compteurs C_0 et C_1 . La première bascule à l'état 1, par exemple la bascule F_2 , fournira un signal R_2 au compteur C_2 pour l'incrémenter. Ce
 35 signal sera également transmis par la chaîne de portes OU à toutes les bascules suivantes et incrémentera les compteurs C_3 à

C_{N-1} . Les sorties des multiplexeurs MUX_2 à MUX_{N-1} seront donc toutes décalées d'un pas. Simultanément, toutes les portes ET 12 qui suivent la bascule F_2 auront une première entrée validée et leur sortie passera à l'état de la deuxième entrée de la porte ET 12 correspondante, c'est-à-dire à l'état de la bascule suivante, la dernière bascule F_{N+k-1} prenant l'état 0. Ainsi, si la bascule F_3 n'était pas à 1, la bascule F_2 prend l'état 0.

Au coup d'horloge suivant, c'est la première bascule F_1 à 1 qui enverra un signal R_1 aux compteurs C_1 à C_{N-1} et qui décalera les sorties du multiplexeur MUX_1 et de tous les multiplexeurs suivants. A nouveau, les contenus de toutes les bascules F_1 à F_{N+k-1} seront décalés d'un pas vers la droite et le cycle se reproduit.

En autant de coups d'horloge qu'il y a de 1 dans les bascules F_0 à F_{N+k-1} , les multiplexeurs MUX_0 à MUX_{N-1} seront mis dans la configuration appropriée par les compteurs C_0 à C_{N-1} .

Avec le schéma de la figure 2, les équations (1) et (2) ci-dessus sont mises en oeuvre par une cascade de portes OU en série. Il peut en résulter des retards relativement importants, ce qui peut entraîner une durée relativement longue pour la phase de réparation. On notera toutefois que ceci n'a d'impact ni sur la durée de la phase de test ni sur la durée ultérieure de fonctionnement normal de la mémoire. Ainsi, on peut utiliser une horloge de fréquence basse spécifique pour la phase de réparation ou utiliser le même signal d'horloge que sur les autres phases mais réduire la fréquence de l'horloge pendant la phase de réparation.

On pourrait aussi mettre en oeuvre les équations précédentes en utilisant un circuit à plus faible retard. La figure 3 représente un exemple de circuit utilisable pour fournir huit signaux R_0 à R_7 à partir de huit états de bascule F_0 à F_7 .

On notera en outre que le signal R_{N+k-1} fourni par la dernière porte OU 11 associée à la dernière bascule F_{N+k-1} est en théorie inutile puisqu'il n'est pas destiné à commander un multiplexeur (il y a seulement N multiplexeurs). Toutefois, ce

signal pourra être utilisé pour indiquer la réussite ou l'échec du processus de réparation. Si $R_{N+k-1}=0$ après k cycles d'horloge de la phase de réparation, la réparation a été couronnée de succès. Par contre, si $R_{N+k-1}=1$ après k cycles d'horloge, la
5 réparation a échoué car cela signifie qu'il y avait plus de k blocs en défaut. R_{N+k-1} peut également être utilisé comme signal d'achèvement de la phase de réparation. C'est-à-dire que l'on n'a pas besoin d'attendre k cycles d'horloge pour parachever la phase de réparation. On peut s'arrêter dès que R_{N+k-1} devient
10 égal à 0. Un des signaux R_{N-1} à R_{N+k} peut être utilisé à cet effet. Le signal R_{N-1} est plus intéressant car sa valeur 0 indiquera plus tôt que les autres la fin de la réparation.

Par ailleurs, la présente invention est susceptible de diverses variantes qui apparaîtront à l'homme de l'art. Par
15 exemple, les équations (1) et (2) peuvent être simplifiées si l'on considère que les blocs redondants U_N à U_{N+k-1} sont exempts de défauts. Dans ce cas, on peut éliminer les bascules F_N à F_{N+k-1} associées à ces blocs et les circuits logiques associés.

On peut réduire la complexité du circuit en faisant
20 certaines suppositions. Par exemple, si l'on suppose que quand on arrive à un module de rang i , i étant nettement inférieur à N , il existe inévitablement un nombre de défauts $k_i < k$, on peut réduire le nombre de sorties des multiplexeurs correspondants qui, au lieu d'être des multiplexeurs à k sorties pourront être
25 des multiplexeurs à k_i sorties. On pourra parallèlement simplifier les compteurs associés à ces multiplexeurs.

Diverses autres simplifications apparaîtront à l'homme de l'art en fonction d'hypothèses pratiques qu'il pourra effectuer sur la réparation de défauts possibles.

30 2.2 Reconfiguration combinatoire

Le circuit décrit précédemment est un circuit à fonctionnement séquentiel.

On cherche maintenant à déterminer directement le décalage à conférer à chaque multiplexeur en fonction de l'état
35 de diverses bascules F_0 à F_{N+k-1} , par exemple par itération.

Pour fixer l'état d'un multiplexeur, on considère que chaque multiplexeur MUX_i a k entrées de commande M^i_0 à M^i_k qui doivent toutes être à zéro sauf l'entrée qui correspond au décalage que l'on veut que le multiplexeur adopte. Par exemple, si
 5 le bloc U_0 est sans défaut, le multiplexeur MUX_0 doit ne provoquer aucun décalage, c'est-à-dire que son entrée de commande M^0_0 doit être à 1 et que toutes ses autres entrées de commande M^1_0 à M^k_0 doivent être à zéro.

En appelant F_i l'état de la bascule F_i et \bar{F}_i l'état complémentaire, les entrées du multiplexeur MUX_0 s'expriment en
 10 fonction de l'état des blocs U_0 à U_k associés comme suit :

$$\begin{aligned} M^0_0 &= \bar{F}_0 \\ M^0_1 &= \bar{F}_1 \cdot F_0 \\ &\dots \\ 15 \quad M^0_k &= \bar{F}_k \cdot F_{k-1} \dots F_0. \end{aligned} \quad (3)$$

Dans ces équations, les points désignent des fonctions ETO. Ces équations se comprennent bien en notant que, si par exemple les modules U_0 et U_1 sont en défaut, le multiplexeur MUX_0 doit effectuer deux décalages, c'est-à-dire que M^0_0 et M^0_1
 20 seront à zéro, que M^0_2 sera à 1 et que tous les M^0_3 à M^0_k seront à zéro.

De façon générale, on peut exprimer l'état de chacune des entrées M^{i+1}_j du multiplexeur MUX_{i+1} en fonction des entrées du multiplexeur MUX_i de la façon suivante :

$$\begin{aligned} 25 \quad M^{i+1}_j &= \bar{F}_{i+j+1} (M^i_j + M^i_{j-1} \cdot F_{i+j} + M^i_{j-2} \cdot F_{i+j-1} \cdot F_{i+j} + \dots \\ &\quad + M^i_0 \cdot F_{i+1} \cdot F_{i+2} \dots F_{i+j}), \quad \text{pour } 0 \leq j \leq k. \end{aligned} \quad (4)$$

A partir des équations (3) et (4), on peut obtenir de façon générale les équations de M^i_j pour toute valeur de i et de j en fonction des variables F_i . Ceci peut être effectué par un
 30 algorithme simple. Un tel algorithme établit d'abord dans l'équation (4) la valeur $i+1=1$ pour obtenir les équations des signaux M^1_j en fonction des variables F_i et des variables M^0_r . Ensuite, il remplace dans les équations résultantes les variables M^0_r par leurs expressions en fonction des variables F_i
 35 comme cela est décrit par les équations (3). Alors, on peut calculer les fonctions M^2_j de façon similaire en fixant $i+1=2$

dans l'équation (4) puis en remplaçant les expressions des fonctions M^0_r et M^1_q obtenues à l'étape précédente et ainsi de suite dans les fonctions M^3_j , M^4_j , ... M^{N-1}_j . Le calcul s'arrête bien entendu pour la position N puisque, comme on l'a vu précédemment, au-delà de cette position, il n'y a pas de multiplexeurs. Le signal $R_f = M^N_0.M^N_1. \dots M^N_k$ peut être utilisé pour signaler la réussite ($R_f=1$) ou l'échec ($R_f=0$) d'une réparation.

3. REPARATION DISTANTE

Comme on l'a rappelé dans l'introduction de la présente demande, on distingue les opérations de réparation de proche en proche telles qu'illustrées en figure 1A des opérations de réparation distante telles qu'illustrées en figure 1B dans laquelle les modules bons restent associés à leurs entrées initiales tandis que les modules en défaut sont remplacés par des modules choisis dans le groupe des modules redondants.

Pour ce mode de réparation distante, on distinguera comme précédemment un mode de réparation séquentiel et un mode de réparation combinatoire.

3.1. Reconfiguration séquentielle

La figure 4 illustre un schéma d'un mode de réalisation d'un système de réparation séquentiel.

Comme dans le schéma de la figure 2, on trouve des multiplexeurs MUX_0 à MUX_{N-1} associés à des entrées d_0 à d_{N-1} destinées à être connectées à des modules non en défaut.

Dans cet exemple, les multiplexeurs sont disposés de droite à gauche au lieu d'être disposés de gauche à droite comme précédemment. On aurait également pu choisir de modifier la numérotation et de partir de N-1 pour aller vers 0 au lieu de partir de 0 pour aller à N-1.

A chaque module est associé un circuit de commande qui au lieu d'être comme précédemment un compteur est cette fois-ci constitué d'un ensemble de $k+1$ bascules pour mémoriser les valeurs exactes des signaux de commande dans un code de 1 parmi $k+1$. Ces circuits de commande sont désignés en figure 4 par les références B_0 à B_{N-1} .

Les bascules indiquant l'état bon ou mauvais des blocs initialement associés aux divers multiplexeurs sont désignées comme précédemment par les références F_0 à F_{N-1} pour les bascules associées aux blocs de base et par les références S_1 à S_k dans l'exemple représenté pour les blocs redondants.

En ce qui concerne les bascules de commande des multiplexeurs, on pourrait également utiliser un nombre plus faible de bascules dans chaque ensemble pour coder les valeurs de ces signaux en un code plus compact, par exemple un code binaire, combiné à un décodeur qui produit le code de 1 parmi $k+1$. Cet ensemble de bascules peut être constitué de façon à réaliser seulement la fonction de charge. Dans ce cas, on devra utiliser un autre circuit pour calculer le nombre de positions dont la borne d'entrée et de sortie d_i doit être décalée vers les blocs redondants et charger ces résultats dans les bascules de réparation.

Une autre possibilité est de prévoir les bascules de réparation selon une configuration de compteur comme on l'avait fait précédemment.

On notera que dans le cas de la figure 4, on effectue la réparation selon un décalage vers la droite ce qui signifie que l'on répare d'abord le module fonctionnel le plus à gauche. On va utiliser deux fonctions séquentielles agissant au rythme d'une horloge qui, à chaque cycle, active notamment les bascules F_0 à F_{N-1} et S_1 à S_k .

La première séquence utilisant les bascules F_0 à F_{N-1} produit des signaux EN_0 à EN_{N-1} qui valident ou non les circuits de commande correspondants. Chaque bascule F_i est reliée au circuit de commande B_i par une porte ET 21 qui reçoit également l'inverse de la fonction OU de toutes les bascules précédentes. Ainsi, si une bascule F_i contient un 0 (bloc correspondant bon), le circuit de commande B_i est inhibé et le multiplexeur MUX_i n'est pas décalé. La sortie de la bascule F_i est reliée par une porte ET 22 à l'entrée de cette bascule F_i . Ainsi, si la bascule F_i contient initialement un 0, son état reste toujours à 0. La deuxième entrée de la porte ET 22 reçoit la fonction OU de

toutes les bascules précédentes. Ainsi, lors d'un cycle d'horloge, seul le circuit de commande associé au bloc en défaut non encore réparé, situé le plus à gauche, est activé.

La seconde séquence fonctionnelle est mise en oeuvre en utilisant les bascules S_1 à S_k et produit à tout instant le nombre de positions dont le bloc en défaut, au cycle considéré, signalé par un signal EN_i actif, doit être décalé vers l'un des blocs redondants (non représentés en figure 4). La sortie de chaque bascule S_i est connectée à son entrée par une porte OU 23 dont la deuxième entrée est connectée à la sortie d'une porte ET 24 qui reçoit les sorties de la bascule S_{i-1} et de la porte ET précédente. On appelle L_i la sortie de la porte ET qui suit la bascule S_i . Les sorties L_i sont fournies à un compteur de 1 combinatoire 25 qui reçoit un 1 sur sa première entrée. Ainsi, le compteur 25 fournit le nombre de positions dont le bloc de base en défaut indiqué par le signal EN_i actif doit être décalé dans les blocs redondants pour réaliser une réparation.

Un compteur de 1 combinatoire peut être réalisé de diverses façons. Deux exemples de telles réalisations pour $k=3$ et $k=5$ en utilisant des additionneurs complets (à trois entrées) FA et des demi-additionneurs (à deux entrées) HA sont représentés en figure 5. Ces compteurs fournissent un résultat codé en binaire. D'autres réalisations peuvent fournir un résultat codé d'une autre manière, par exemple selon un code de 1 parmi $k+1$.

Un compteur de 1 séquentiel pourrait également être mis en oeuvre. Par exemple ceci peut être réalisé en utilisant un circuit de décalage à $k+1$ cellules. La cellule 0 est chargée de 1. Les signaux L_1, L_2, \dots, L_k sont chargés dans les cellules suivantes. Les contenus seront décalés vers la droite jusqu'à ce que le premier 0 soit fourni vers la sortie.

Pour le calcul des décalages, on utilisera des fonctions intermédiaires FI_i . La fonction FI_i est à 1 si l'un quelconque des signaux $F_i, F_{i+1} \dots F_{N-1}$ est à 1. Ceci donne les équations récursives suivantes :

$$FI_i = FI_{i-1} + F_i, \quad \text{avec } 0 \leq i \leq N-1 \text{ et } FI_{-1}=0 \quad (5)$$

La bascule F_i maintient son état tant que l'une des bascules $F_{i-1}, F_{i-2}, \dots, F_{N-1}$ est à 1. Cette condition implique que $FI_{i-1} = 1$. On obtient ainsi :

$$F_i(t+1) = FI_{i-1} \cdot F_i(t), \quad \text{avec } 0 \leq i \leq N-1 \quad (6)$$

5 Le signal EN_i passe à 1 si F_i est à 1 et que toutes les bascules F_{i-1} à F_0 sont à 0. Ceci implique que $FI_{i-1} = 0$. On obtient ainsi :

$$EN_i = \overline{FI_{i-1}} \cdot F_i \quad (7)$$

10 Le circuit formé à partir des bascules S_i met à jour le contenu de ces bascules à chaque cycle d'horloge. Cette mise à jour est telle que à tout cycle les bascules S_i indiquant un module disponible contiennent un 0 et que les autres bascules S_i contiennent un 1. Ainsi un bloc redondant disponible est un bloc exempt de défauts qui n'est pas encore occupé par la réparation
15 d'un bloc fonctionnel en défaut. Un bloc redondant en défaut reste toujours indisponible, c'est-à-dire que S_i reste toujours égal à 1. Un bloc redondant exempt de défaut devient indisponible à un cycle $t+1$ si au cycle t toutes les bascules S_i précédentes sont à 1. Ainsi on a :

$$20 \quad L_0 = 1, \quad L_i = L_{i-1} \cdot S_i, \quad 1 \leq i \leq k \quad (8)$$

$$S_i(t+1) = S_i(t) + L_{i-1}, \quad 1 \leq i \leq k \quad (9)$$

D'après ces équations, les signaux L_1, L_2, \dots, L_k prennent les valeurs $L_1=L_2=\dots=L_{j-1}=1, L_j=L_{j+1}=\dots=L_k=0$, où j est le plus petit entier pour lequel $S_j = 0$. Dans ce cas, le
25 bloc redondant $j+1$ est le premier bloc redondant disponible pour réaliser une réparation.

Les signaux L_k et FI_{N-1} peuvent être utilisés pour achever la phase de réparation. $FI_{N-1}=0$ indique que toutes les positions sont correctes et que la réparation peut s'achever.
30 $L_k=1$ indique qu'il n'y a plus de module redondant disponible.

La réparation peut également être réalisée en utilisant des compteurs de réparation plutôt que des bascules de réparation B_i . Le compteur de réparation peut être un compteur binaire suivi d'un décodeur pour produire un code de 1 parmi
35 $k+1$, ou un registre à décalage. Dans ce cas, le compteur 25 peut être remplacé par un compteur séquentiel de 1 tel qu'un circuit

de décalage. La valeur 1 de la cellule la plus à gauche de ce circuit de décalage sera utilisée pour conserver les états des bascules F et S. La valeur 0 de cette cellule sera utilisée pour maintenir l'état du compteur séquentiel jusqu'à son chargement
5 suivant.

La cellule la plus à gauche du circuit de décalage sera à 1 pendant k_i cycles d'horloge, où k_i est égal au nombre de signaux L_j ayant la valeur 1, plus 1. Alors, k_i est égal au nombre de positions dont le bloc U_i en défaut indiqué par l'état
10 $EN_i=1$ doit être décalé vers les modules redondants. Le signal EN_i sera maintenu à la valeur 1 pendant k_i cycles d'horloge. Ainsi, en utilisant le signal EN_i en tant que signal qui valide l'opération de comptage du compteur de réparation de position i , on pourra produire les valeurs requises sur les signaux de
15 commande du multiplexeur de réparation de position i .

Avec cette dernière mise en oeuvre, on peut également charger dans le registre mettant en oeuvre le compteur de 1 les signaux $S_1, S_2 \dots S_k$ au lieu des signaux $L_1, L_2 \dots L_k$. La raison en est que ces signaux sont égaux jusqu'à et y compris la position
20 la plus à gauche dont l'état est 0. Ces positions sont celles qui déterminent le fonctionnement du circuit.

3.2 Reconfiguration combinatoire

Comme précédemment, le système de réparation à distance peut également être réalisé par une fonction combinatoire que l'on définira de façon incrémentielle. En reprenant
25 les notations de la section 2.2, on a les équations :

$$\begin{aligned} M^0_0 &= \bar{F}_0 \\ M^0_1 &= \bar{S}_1.F_0 \\ &\dots \\ M^0_k &= \bar{S}_k.S_{k-1} \dots S_1.F_0. \end{aligned} \tag{10}$$

Les équations correspondant à une borne arbitraire d_j impliquent les variables F et S. Pour calculer les variables M^{i+1}_j on ne peut utiliser les variables M^i_r comme indicateurs du nombre de blocs redondants occupés après la réparation de certains blocs de base. Ceci est dû au fait que si le bloc de base
35 U_i est exempt de défaut, toutes les fonctions M^i_1 à M^i_k sont à 0

bien que certains des blocs U_0 à U_{i-1} puissent être en défaut. Ainsi, on introduit des variables intermédiaires MI^i_j qui comptent le nombre de blocs redondants occupés par les blocs de base en défaut. Ces variables sont déterminées par les équations suivantes.

Pour la position 0, les variables MI sont égales aux variables M :

$$MI^0_j = M^0_j \text{ pour } 0 \leq j \leq k \quad (11)$$

Pour la position $i+1$, les variables MI^{i+1}_j sont égales aux variables MI^i_j si le bloc U_{i+1} est exempt de défauts. Sinon elles sont égales à M^{i+1}_j . Ainsi, on a :

$$MI^{i+1}_j = MI^i_j \cdot \bar{F}_{i+1} + M^{i+1}_j \cdot F_{i+1} \quad 0 \leq i \leq N-2 \text{ et } 0 \leq j \leq k \quad (12)$$

La variable M^{i+1}_0 est égale à 1 si le bloc U_{i+1} est exempt de défaut. La variable M^{i+1}_{j+1} est égale à 0 si le bloc U_{i+1} est exempt de défauts. Si U_{i+1} est en défaut, M^{i+1}_{j+1} devient égal à 1 si S_{j+1} est exempt de défauts et s'il existe un entier $r \leq j$ tel que r blocs redondants en défaut et exempts de défauts sont déjà occupés pour réparer les modules U_q avec $q \leq i$, et les blocs redondants $S_{r+1}, S_{r+2} \dots S_j$ sont en défauts. Cette analyse conduit aux équations :

$$\begin{aligned} M^{i+1}_0 &= \bar{F}_{i+1}, \\ M^{i+1}_{j+1} &= \bar{S}_{j+1} F_{i+1} (MI^i_j + MI^i_{j-1} S_j + MI^i_{j-2} S_{j-1} S_j + \dots \\ &\quad + MI^i_0 S_1 S_2 \dots S_j), \text{ pour } 0 \leq j \leq k-1, 0 \leq i \leq k-2, \end{aligned} \quad (13)$$

Ainsi, les équations (10) à (13) décrivent la fonction M de façon récursive et peuvent être mises en oeuvre par tout logiciel ou matériel approprié.

Bien entendu, la présente invention est susceptible de diverses variantes et modifications qui apparaîtront à l'homme de l'art. En particulier, chaque bascule d'indication d'état est associée à des signaux de validation pour, dans un mode de test se charger en fonction de l'état bon ou mauvais du bloc correspondant, et, en mode de réparation, fonctionner de la façon décrite précédemment.

35 Variantes de multiplexage

En figure 2 les données peuvent se propager :

- des modules U_1 à U_{n+k-1} vers les bornes d_0 à d_{n+k-1} (par exemple dans le cas où d_0 à d_{n+k-1} sont le bus de données en lecture d'une mémoire). On appellera ce cas configuration de type lecture.
- 5 - des bornes d_0 à d_{n+k-1} vers les modules U_1 à U_{n+k-1} (par exemple dans le cas où d_0 à d_{n+k-1} sont le bus des données en écriture d'une mémoire, ou dans le cas où d_0 à d_{n+k-1} sont les signaux de sélection des modules). On appellera ce cas configuration de type écriture.
- 10 - dans les deux directions (par exemple dans le cas où d_0 à d_{n+k-1} sont le bus bidirectionnel de lecture/écriture d'une mémoire). On appellera ce cas configuration bidirectionnelle.

Dans le cas de la configuration bidirectionnelle, les multiplexeurs utilisés doivent permettre une propagation
15 bidirectionnelle des signaux. Un multiplexeur du type 1-parmi-4 est présenté en figure 6. Un multiplexeur bidirectionnel peut être également utilisé dans le cas des configurations du type lecture et du type écriture.

Dans le cas de la configuration du type lecture, on
20 peut aussi utiliser un multiplexeur permettant uniquement la propagation des k signaux D_0 à D_{k-1} de sortie des modules U_1 à U_{n+k-1} vers le signal d . Un tel multiplexeur pour le cas $k=4$ est présenté en figure 7.

Dans le cas de la configuration du type écriture, on
25 peut aussi utiliser un "multiplexeur" permettant uniquement la propagation du signal d vers les k signaux D_0 à D_{k-1} . Ce circuit, qui n'est pas un multiplexeur au sens habituel du terme, sera appelé MUX k -parmi-1. Un circuit de ce type pour le cas $k=4$ est représenté en figure 8. On remarque que dans cette figure $M_j^i=1$
30 implique $D_j=0$. Par conséquent, si on doit générer un signal G en combinant plusieurs signaux D_j en provenance des sorties de plusieurs circuits de type k -parmi-1, il faut utiliser une porte OU recevant les signaux D_j et fournissant le signal G . Par contre, si on utilise un multiplexeur bidirectionnel, réalisé
35 comme dans la figure 6, la porte OU ne sera pas utilisée, car le signal G sera généré en utilisant une connexion câblée des

signaux D_j . Ainsi, une configuration de type écriture, utilisant des circuits de type MUX k-parmi-1, nécessitera de remplacer la connexion câblée des signaux D_j par des portes OU, tandis que l'utilisation de multiplexeurs bidirectionnels ne nécessite pas l'utilisation de portes OU. Néanmoins, dans certains cas, le signal fourni à un module doit prendre la valeur 1 à l'état actif et la valeur 0 à l'état inactif. Cette contrainte n'est pas satisfaite quand on utilise le multiplexeur bidirectionnel de la figure 6 ; par contre elle est satisfaite quand on utilise le circuit de type MUX k-parmi-1 présenté dans la figure 8, combiné avec la porte OU. Il se peut aussi que le signal fourni à un module doive prendre la valeur 0 à l'état actif, et la valeur 1 à l'état inactif. Dans ce cas on utilisera des circuits de type MUX k-parmi-1 réalisant la fonction $M_j^i=1$ implique $D_i=d$ et $M_j^i=0$ implique $D_i=1$, combinés avec des portes ET.

Dans d'autres implémentations, il se peut qu'on combine les sorties d des plusieurs multiplexeurs 1-parmi-k pour générer un signal G. Si on utilise des multiplexeurs bidirectionnels, on peut générer le signal G en utilisant une connexion câblée des signaux d. Par contre, si on utilise des multiplexeurs 1-parmi-k, réalisés comme dans la figure 7, la connexion câblée doit être remplacée par une porte OU. La raison est que le multiplexeur de la figure 7 réalise la fonction : $M_0^i, M_1^i, \dots, M_j^i, \dots, M_k^i = 00\dots 1\dots 0$ implique $d=D_j$ et $M_0^i=M_1^i, \dots, M_j^i, \dots, M_k^i=0$ implique $d=0$. Ainsi, il sera moins coûteux d'utiliser un multiplexeur bidirectionnel. Néanmoins, il se peut que le signal G prenne la valeur 0 à l'état inactif. Dans ce cas, il sera nécessaire d'utiliser un multiplexeur réalisé comme dans la figure 7. D'autre part, si le signal G doit prendre la valeur 1 à l'état inactif, on doit utiliser des multiplexeurs de type 1-parmi-k, réalisant la fonction : $M_0^i, M_1^i, \dots, M_j^i, \dots, M_k^i = 00\dots 1\dots 0$ implique $d=D_j$, et $M_0^i = M_1^i, \dots, M_j^i, \dots, M_k^i = 0$ implique $d=1$.

Un dernier cas des circuits de multiplexage correspond à des circuits réalisant les mêmes fonctions que celles décrites précédemment, à l'exception près qu'on applique des valeurs

complémentaires à leurs entrées. Par exemple, l'implication $M_0^i, M_1^i, \dots, M_j^i, \dots, M_k^i = 00\dots 1\dots 0$ implique $d=D_j$, devient $M_0^i, M_1^i, \dots, M_j^i, \dots, M_k^i = 11\dots 0\dots 1$ implique $d=D_j$.

Une autre façon de réaliser l'aiguillage de proche en proche est représentée en figure 9, dans laquelle $N=3$ et $k=2$. Pour réaliser le multiplexage, on utilise des circuits d'aiguillage, tels que présentés dans les figures 6, 7, et 8. Si les circuits de la figure 1A sont des multiplexeurs bidirectionnels, la figure 9 utilisera des multiplexeurs bidirectionnels. Si les circuits de la figure 1A sont des multiplexeurs 1-parmi- k , comme le circuit de la figure 7, la figure 9 utilisera des circuits d'aiguillage du type MUX r -parmi-1, comme le circuit de la figure 8. Si les circuits de la figure 1A sont des circuits du type MUX k -parmi-1, comme le circuit de la figure 8, la figure 9 utilisera des multiplexeurs 1-parmi- r , comme le circuit de la figure 7.

On considère que la figure 9 utilise des circuits du type MUX r -parmi-1. La borne d de chacun de ces circuits est connectée sur le signal d'entrée/sortie d'un des modules $U_1, U_2, \dots, U_{N+k-1}$. On appellera CM_i le circuit d'aiguillage connecté à l'entrée/sortie du module U_i .

Les bornes D_j des circuits d'aiguillage sont combinées pour générer les signaux $d_0, d_1, \dots, d_{N+k-1}$. Pour les circuits d'aiguillage CM_i du type r -parmi-1, les valeurs de r seront :

25 $r = i + 1, \quad \text{pour } 0 \leq i \leq k-1,$
 $r = k + 1, \quad \text{pour } k \leq i \leq N-1,$
 $r = N + k - i, \text{ pour } N \leq i \leq N+k-1.$

Pour $0 \leq i+j \leq N-1$, le signal M_j^i sélectionne la connexion de la borne D_i du circuit CM_{i+j} à la borne d'entrée/sortie du module UM_{i+j} . Pour $N \leq i+j \leq N+k-1$, le signal M_j^i sélectionne la connexion de la borne D_{N-j-1} du circuit CM_{i+j} à la borne d'entrée/sortie du module UM_{i+j} . Les bornes D_i sélectionnés par les signaux $M_0^i, M_1^i, \dots, M_k^i$ sont combinées pour générer le signal d_i . La connectique sera exactement la même si les circuits d'aiguillage de la figure 9 étaient des multiplexeurs 1-parmi- r .

Une transformation similaire peut être utilisée dans le cas du système de configuration distante illustré dans la figure 1B. Dans ce cas le paramètre r des circuits d'aiguillage prend les valeurs suivantes :

- 5 $r = 1$, pour $0 \leq i \leq N-1$, et
 $r = k + 1$, pour $N \leq i \leq N+k-1$.

Pour $0 \leq i \leq N-1$, le signal M_i^t sélectionne la connexion de la borne d'entrée/sortie du module U_i à la borne D_0 du circuit d'aiguillage CM_i , qui a en effet une seule borne du type D_i . Pour
 10 $0 \leq i \leq N-1$, et $1 \leq j \leq k$, le signal M_j^t sélectionne la connexion de la borne d'entrée/sortie du module U_{N+j-1} à la borne D_i du circuit d'aiguillage CM_{N+j-1} . Les bornes D_i sélectionnées par les signaux $M_0^t, M_1^t, \dots, M_k^t$ sont combinées pour générer le signal d_i .

Dans les systèmes obtenus par les transformations précédentes, les signaux M_j^t prennent les mêmes valeurs que dans le
 15 système d'origine. Les solutions de génération des signaux M_j^t que nous avons présentées sont donc valables pour les systèmes obtenus par ces transformations.

Réparation utilisant une redondance insuffisante

20 On considère maintenant un système incluant N modules fonctionnels, k modules redondants et capables de réparer jusqu'à r modules défectueux, avec $r > k \geq 0$. Dans le cas où le nombre de modules défectueux ne dépasse pas k , le système, après réparation, fonctionnera avec N modules. Si le nombre de
 25 modules défectueux est q avec $k < q \leq r$, le système fonctionnera avec $N+k-q$ modules. On suppose qu'on dispose de N signaux de sélection d_0, d_1, \dots, d_{N-1} , chacun sélectionnant un module. La réparation consistera à rerouter les signaux $d_0, d_1, \dots, d_{N+k-q-1}$ de façon à remplacer chaque module fonctionnel défectueux par un
 30 module sans faute. De plus, le système de réparation pourra éventuellement fournir des signaux indiquant la réduction du nombre de modules de façon que le reste du système utilise seulement les ressources des modules non défectueux qui seront vus par le système comme étant les $N+k-q$ premiers modules de la
 35 configuration.

Pour illustrer l'intérêt de cette solution, considérons que les modules $U_0, U_1, \dots, U_{N+k-1}$, sont des blocs d'une mémoire, chacun incluant R adresses. Le système peut exploiter une mémoire incluant un total de RN adresses. Dans cette configuration, il offre les meilleures performances, mais il pourra aussi fonctionner avec une mémoire de capacité plus faible.

Les signaux d_0, d_1, \dots, d_{N-1} sont générés par exemple, par le décodage d'un nombre de bits d'adressage égal à $\lceil \log_2(N) \rceil$. La valeur 00...0 de ces bits active le signal d_0 , la valeur 00...1 active le signal d_1 , etc., jusqu'à la valeur $N-1$ exprimée en binaire qui activera d_{N-1} . Dans le cas de q modules défaillants où le système utilisera un espace mémoire incluant seulement $R(N+k-q)$ adresses, il est important que cet espace d'adressage ne soit pas fragmenté, c'est-à-dire que le système dispose d'une mémoire couvrant l'espace d'adressage 0, 1, ... $R(N+k-q)$. Le système de réparation proposé, permet cette unification par le reroutage de signaux $d_0, d_1, \dots, d_{N+k-q-1}$ et fournit éventuellement des signaux indiquant l'espace d'adressage invalide.

La technique que nous proposons pour faire cette réparation implémente le même circuit de réparation dans le cas où le nombre k des modules redondants est égal au nombre r de modules défaillants que nous envisageons de réparer, à l'exception près de certains multiplexeurs, comme cela sera décrit dans ce qui suit.

Nous considérons ici une réparation locale. Les modules manquants pour avoir un nombre de modules redondants égal à r sont les modules U_{N+r-i} tels que $1 \leq i \leq r-k$.

Les groupes des signaux de contrôle qui sélectionnent tout module U_{N+r-i} tel que $1 \leq i \leq r-k$ sont les signaux M^{N-j}_{r+j-i} tels que $1 \leq i \leq r-k$ et $1 \leq j \leq i$. Ce groupe de signaux de contrôle ne rentrent pas dans le multiplexeur de rang $N-j$, qui aura ainsi $r+j-i$ entrées de contrôle au lieu de $r+1$. Ce multiplexeur aura par ailleurs $r+j-i$ bornes de type D_q au lieu de $r+1$. Les bornes manquantes seront les bornes D_{r-j+1} tels que $1 \leq j \leq i$.

Par ailleurs, les signaux de contrôle non utilisés par les multiplexeurs pourront servir pour indiquer l'espace d'adressage invalide. Pour ce faire, on formera les signaux binaires $@i$ pour tout i , tel que $1 \leq i \leq r-k$. Ces signaux sont

5 déterminés par :

$$@i = \prod_{j=1}^i M^{N-j}_{r+j-i}$$

Dans cette formule, $\prod_{j=1}^i G_j$ représente le ET logique de tous les termes G_j tels que $1 \leq j \leq i$. Le signal $@i$ au plus petit indice i tel que $@i=1$ indique que l'espace d'adressage est limité aux seules

10 $(N+k+i-r-1)R$ premières adresses.

La figure 10 donne l'exemple de la technique pour $N=4$ modules fonctionnels, $k=2$ modules redondants et $r=3$ modules défaillants réparables. On observe que l'implémentation est la même que dans le cas $N=4$, $k=r=3$ (4 modules fonctionnels, 3

15 modules redondants et 3 modules défaillants réparables), avec l'exception du dernier multiplexeur qui est un multiplexeur 1-parmi-3 au lieu de 1-parmi-4. Le signal $@1=M^3$, indique que les R dernières adresses sont invalides.

Réparation pour mémoire CACHE

20 Une dernière solution de réparation amenant à des coûts très faibles, concerne le cas des mémoires CACHE. Ces mémoires peuvent disposer d'un point de mémorisation ou flag, appelé ci-après flag de validité ou F_g . F_g indique si un mot mémoire est valide ou invalide. Par exemple, $F_g=1$ indique que le

25 mot est valide et $F_g=0$ indique que le mot est invalide. Dans ce dernier cas, le système effectue l'opération dans une autre de ses mémoires. La valeur du point de mémorisation est manipulée pendant le fonctionnement du système, permettant de gérer la cohérence des accès aux mémoires CACHE et aux autres mémoires.

30 Nous proposons ici d'introduire pour chaque mot de la mémoire CACHE (ou pour un ensemble de mots) un autre point de mémorisation du type flag, appelé ci-après flag de défaut ou F_d . Le contenu de F_d est manipulé uniquement durant le test du circuit, et il est fixé à la valeur indiquant un mot défaillant (par

35 exemple pour $F_d=1$) ou à la valeur indiquant un mot non défaillant (par exemple pour $F_d=0$). Lors du fonctionnement du système,

le contenu de F_d est bloqué à sa valeur fixée lors de la phase du test. Ensuite, si F_d indique un mot défaillant, l'opération s'effectuera dans une autre mémoire. Etant donné que la valeur de F_d a été fixée avant que l'opération du système débute, aucune opération ne sera effectuée dans le mot défaillant, et toute donnée qui aurait pu être stockée dans le mot s'il avait été non défaillant, sera stockée dans une autre mémoire, permettant un fonctionnement cohérent du système. Etant donné que le nombre des défaillances dans un circuit intégré est généralement faible, cette approche réduit de façon non significative la capacité de la CACHE et elle n'a qu'une influence minime sur les performances du système.

Les mémoires CACHE sont utilisées pour réduire la fréquence des accès dans la mémoire du système. Les cellules d'un mot d'une mémoire CACHE sont partagées dans une partie appelée clé et dans une partie de données. Les bits d'adressage de la mémoire du système sont partagés en deux parties, ci-après appelées première partie et deuxième partie. La première partie des bits d'adressage est utilisée pour sélectionner un mot de la mémoire CACHE. Dans une opération d'écriture, la deuxième partie des bits d'adressage est stockée dans la partie clé du mot sélectionné, tandis que les données de l'opération d'écriture sont stockées dans la partie de données. Dans une opération de lecture, la partie clé et la partie de données du mot sélectionné sont lues. Le contenu de la partie clé est comparé à la deuxième partie des bits d'adressage de l'opération écriture. Si la comparaison réussit, on dit que l'accès à la mémoire CACHE est réussi, et on utilise les valeurs lues dans la partie données comme résultat de l'opération de lecture. Si la comparaison échoue, on accède à la mémoire du système pour effectuer la lecture. Dans la plupart des cas, chaque valeur de la première partie des bits d'adressage sélectionne un seul mot de la mémoire CACHE. Mais dans certaines mémoires CACHE la même valeur de la première partie des bits d'adressage peut sélectionner plusieurs mots. Ainsi, dans le premier cas, si un mot est invalide car son flag de défaut indique un mot défaillant,

alors chaque opération, telle que la valeur de la première partie des bits d'adressage sélectionne ce mot, va s'effectuer dans la mémoire du système. Dans le deuxième cas, une opération sélectionnant un mot dont le flag de défaut indique qu'il est
5 défaillant n'accédera pas nécessairement à la mémoire du système, car l'opération peut réussir dans un des autres mots de la mémoire CACHE sélectionné par la même valeur de la première partie des bits d'adressage.

Une technique similaire peut être utilisée pour des
10 mémoires de type CAM (Content Adressable Memory). On peut voir une mémoire CAM comme une mémoire CACHE dont la première partie des bits d'adressage est vide. Ainsi, tous les bits d'adressage sont stockés dans la partie clé. Néanmoins, la partie clé de chaque mot possède un comparateur. Ainsi, dans une opération de
15 lecture, l'adresse de lecture est comparée en parallèle au contenu de la partie clé de chaque mot. Si la comparaison réussit pour un mot, la sortie du comparateur de ce mot sélectionne la partie de données du mot. Ainsi, on peut lire le contenu de la partie de données de ce mot à la sortie de la mémoire.

20 En écriture, il y a deux cas possibles.

- Comme pour une lecture, l'adresse est comparée à la partie clé des mots. Si la comparaison réussit pour un mot de la CAM, la sortie du comparateur de ce mot sélectionne la partie de données du mot et on effectue l'écriture dans cette
25 partie.
- Il n'y a pas de comparaison de l'adresse, mais on utilise un critère pour sélectionner un mot de la CAM, et on écrit l'adresse d'écriture dans la partie clé du mot, et la donnée de l'opération d'écriture dans la partie de données du mot.
30 Dans ce cas, pour sélectionner le mot à écrire, divers critères sont utilisés d'un système à l'autre. Souvent, ce type d'écriture sélectionne le mot auquel on a accédé le moins récemment, mais d'autres critères sont aussi utilisés selon le système. L'utilisation des flags qui permettent ou
35 interdisent l'accès à un mot est aussi une pratique courante.

Pour effectuer la réparation d'une mémoire CAM, on rajoute à chaque mot une cellule de flag de défaut Fd. En phase de test, si un défaut est détecté dans un mot, on écrit 1 dans la cellule du flag Fd du mot. En fonctionnement normal, la valeur 1 du flag Fd interdit tout accès au mot défaillant. Les opérations visant le mot défaillant peuvent alors être effectuées dans une autre mémoire du système. Néanmoins, en contraste avec une mémoire CACHE, dans une mémoire CAM tout mot est remplaçable par tout autre mot. En fait, étant donné que chaque mot est sélectionné uniquement par le contenu de sa partie clé, il n'y a pas de différenciation de mots en fonction de la valeur des bits d'adressage. Ainsi, dans le cas présent, on n'a pas besoin d'accéder à une autre mémoire du système, car on peut effectuer toutes les opérations dans les mots non défaillants de la mémoire CAM réparée. Par ailleurs, la mémoire CAM peut comporter un nombre de mots suffisant de façon à avoir après réparation une capacité satisfaisante pour le système.

La mémoire CAM proposée ici, peut être utilisée en outre pour effectuer la réparation d'une autre mémoire en remplaçant chaque mot défaillant de la mémoire par un mot non défaillant de la mémoire CAM. Pour ce faire, on procède selon les étapes suivantes.

- On teste d'abord la mémoire CAM et, chaque fois qu'on détecte un défaut dans un mot, on écrit 1 dans le flag Fd de ce mot. Le contenu des flags Fd n'est plus modifié après la fin du test de la mémoire CAM.
- On teste la mémoire à réparer et, chaque fois qu'on détecte un défaut dans un mot, on écrit la valeur de l'adresse qui a sélectionné le mot défaillant dans la partie clé d'un mot de la mémoire CAM. Pour sélectionner le mot de la CAM dans lequel on va effectuer cette écriture, on peut utiliser diverses procédures. Une telle procédure consiste à utiliser, pendant la phase du test de la mémoire, un compteur binaire avec un décodeur. Chaque sortie du décodeur sélectionne en phase de test de la mémoire un mot de la CAM. A chaque écriture dans la CAM on incrémente le compteur. Ainsi, la pro-

chaine écriture d'une adresse se fera dans un autre mot. Pendant cette phase, on vérifie aussi la valeur du flag Fd. Si pour un mot de la CAM on trouve $Fd = 1$, on sélectionne un autre mot de la CAM pour effectuer une écriture d'adresse.

5 Dans l'implémentation utilisant le compteur et le décodeur, chaque fois qu'on sélectionne un mot avec $Fd=1$, on incrémente de nouveau le compteur, avant d'effectuer une écriture d'adresse. Bien sûr, le compteur et le décodeur peuvent être

10 remplacés par un registre à décalage dont un bit est initialisé à 1, et tous les autres bits à 0.

- En fonctionnement normal, chaque adresse d'une opération d'écriture ou de lecture dans la mémoire est comparée en parallèle aux parties clés de tous les mots de la CAM. Si la comparaison réussit pour un mot, qui a par ailleurs $Fd=0$, on

15 effectue l'opération courante dans ce mot de la CAM, au lieu de l'effectuer dans la mémoire. Ainsi, dans le cas d'une opération destinée à faire une écriture dans la mémoire, on écrit les données à écrire dans la partie de données de ce mot de la CAM, et en cas de lecture on lit la partie de données

20 de ce mot de la CAM et on fournit le résultat au système au lieu des données qu'on devrait lire dans la mémoire.

On réussit ainsi à réparer les mots défaillants dans la mémoire, tout en se protégeant contre les défauts dans les mots de la CAM. Notons que, si un mot de la CAM contient une

25 faute dans la cellule de flag Fd, et dans une autre partie du mot, le flag défaillant pourrait indiquer que le mot n'est pas défaillant. Pour réduire la probabilité de cet événement, on peut répliquer la cellule de ce flag une ou plusieurs fois, et invalider le mot si une au moins de ces cellules de flag est à

30 1. Néanmoins, la probabilité d'occurrence de tels défauts reste très faible pour les technologies actuelles et on utilisera une seule cellule pour le flag Fd. L'augmentation de la densité des défauts prévue pour les nanotechnologies pourrait apporter un intérêt à la réplification des cellules de flag Fd.

35 Le système de réparation peut être prévu pour qu'on puisse reconfigurer la mémoire non seulement durant le test de

fabrication mais aussi pendant les différentes phases de test exécutées durant la vie du circuit. Dans ce cas, Fd sera stocké dans un point de mémorisation reprogrammable (volatil ou non-volatil). D'autre part, si la réparation s'effectue uniquement pendant le test de fabrication, Fd pourrait être stocké dans un point de mémorisation non reprogrammable, par exemple dans un dispositif fixant une valeur logique à l'aide d'un fusible.

Dans la description précédente des systèmes à reconfiguration combinatoire, on a décrit des codages de type 1 parmi $k+1$ des états M^i_0 à M^i_k des entrées de démultiplexeurs. On aurait également pu prévoir des systèmes à codage binaire. L'homme de métier pourra réaliser une correspondance biunivoque entre ces deux types de codage. Dans ce cas, les signaux binaires prendront la valeur q quand le signal M^i_q est égal à 1 et que tous les autres signaux M^i_0 à M^i_k sont à 0. L'homme du métier pourra par ailleurs appliquer ce principe pour coder le nombre de décalages d'une position selon tout autre code.

REVENDICATIONS

1. Dispositif de reconfiguration d'un ensemble de N modules électroniques de base associés à k modules redondants, k des N+k modules étant susceptibles de présenter un défaut, comprenant :

5 N multiplexeurs de 1 parmi k+1 (MUX_i), chaque multiplexeur ayant une première borne (d_i) susceptible d'être connectée à l'une de k+1 deuxièmes bornes respectivement connectées aux k+1 bornes d'entrée/sortie d'un groupe ordonné de modules composé d'un module de base (U_i) et de k autres modules ;

10 N+k bascules (F_i), chaque bascule étant indicatrice d'un état bon ou mauvais d'un des N modules de base et des k modules dudit groupe ordonné ; et

un moyen logique associé à chaque multiplexeur de rang j, j étant un entier compris entre 0 et N, pour

- 15 - déterminer un premier nombre correspondant au nombre de bascules de rang 0 à j indiquant un état mauvais,
- déterminer un deuxième nombre correspondant au nombre de modules du groupe ordonné associé au module de rang j, à compter, en partant du premier module dudit groupe ordonné, pour
- 20 trouver un nombre de modules bons égal au premier nombre,
- connecter la première borne du multiplexeur à sa deuxième borne de rang égal au deuxième nombre, et
- inhiber éventuellement le décalage des multiplexeurs correspondant à des blocs bons.

25 2. Dispositif de reconfiguration selon la revendication 1, dans lequel la réparation est effectuée de proche en proche et de façon séquentielle, caractérisé en ce que le moyen logique (F_i , 11, 12, C_i) fournit à chaque cycle d'une horloge une commande de décalage éventuel des multiplexeurs et une

30 commande des contenus (F_i) des bascules selon les équations suivantes :

$$R_i = R_{i-1} + F_i, \quad \text{pour } N+k-1 \geq i \geq 0 \quad (1)$$

$$F_i(t+1) = R_i.F_{i+1}(t), \quad \text{pour } N+k-1 \geq i \geq 0 \quad (2)$$

où t et t+1 indiquent deux instants successifs.

3. Dispositif de reconfiguration selon la revendication 1, dans lequel le moyen logique comprend :

une porte OU (11) recevant la sortie de chaque bascule de rang i (F_i) et de la bascule précédente (F_{i-1}) ;

5 une porte ET (12) recevant les sorties de la porte OU (11) et de la bascule de rang $i+1$ (F_{i+1}) ; et

un compteur (C_i) de commande du multiplexeur (MUX_i), recevant la sortie de la porte OU (11)

4. Dispositif de reconfiguration selon la revendication 1, dans lequel la réparation est effectuée de proche en proche et est réalisée selon l'équation combinatoire :

$$M^{i+1}_j = \bar{F}_{i+j+1} (M^{i}_j + M^{i}_{j-1} \cdot F_{i+j} + M^{i}_{j-2} \cdot F_{i+j-1} \cdot F_{i+j} + \dots + M^{i}_0 \cdot F_{i+1} \cdot F_{i+2} \dots F_{i+j}), \quad \text{pour } 0 \leq j \leq k. \quad (4)$$

5. Dispositif de reconfiguration selon la revendication 1, dans lequel la réparation est effectuée de façon distante et de façon séquentielle, caractérisé en ce que le moyen logique (F_i , S_j , 21-25, B_i) fournit à chaque cycle d'une horloge une commande de décalage éventuel d'un multiplexeur en défaut et une commande des contenus (F_i , S_j) des bascules selon les équations suivantes :

$$FI_i = FI_{i-1} + F_i, \quad \text{si } 0 \leq i \leq N-1, \text{ avec } FI_{-1} = 0 \quad (5)$$

$$F_i(t+1) = FI_{i-1} \cdot F_i(t), \quad \text{si } 0 \leq i \leq N-1 \quad (6)$$

$$EN_i = FI_{i-1} \cdot F_i \quad (7)$$

$$L_0 = 1, \quad L_i = L_{i-1} \cdot S_i, \quad 1 \leq i \leq k$$

$$(8)$$

$$S_i(t+1) = S_i(t) + L_{i-1}, \quad 1 \leq i \leq k \quad (9)$$

où : FI_i est une variable,

$F_i(t)$ désigne l'état d'une bascule associée à un module de base à l'instant t ,

30 EN_i désigne un signal de validation appliqué à un circuit de commande de multiplexeur,

L_i désigne un signal tel que la somme des L_i , plus 1, est appliquée à un circuit de commande de multiplexeur à l'instant t ,

$S_i(t)$ désigne l'état d'une bascule associée à un module de remplacement à l'instant t .

6. Dispositif de reconfiguration selon la revendication 1, dans lequel la réparation est effectuée de façon distante et est réalisée selon l'équation combinatoire :

$$M^{i+1}_0 = \bar{F}_{i+1},$$

$$M^{i+1}_{j+1} = \bar{S}_{j+1} F_{i+1} (M^i_j + M^i_{j-1} S_j + M^i_{j-2} S_{j-1} S_j + \dots + M^i_0 S_1 S_2 \dots S_j)$$

pour $0 \leq j \leq k-1$, $0 \leq i \leq k-2$, (13)

où : F_i désigne l'état d'une bascule associée à un module de base,

S_i désigne l'état d'une bascule associée à un module de remplacement.

M^i_j est une variable telle que

$$M^{i+1}_j = M^i_j \cdot \bar{F}_{i+1} + M^{i+1}_j \cdot F_{i+1}$$

$0 \leq i \leq N-2$ et $0 \leq j \leq k$ (12)

1/5

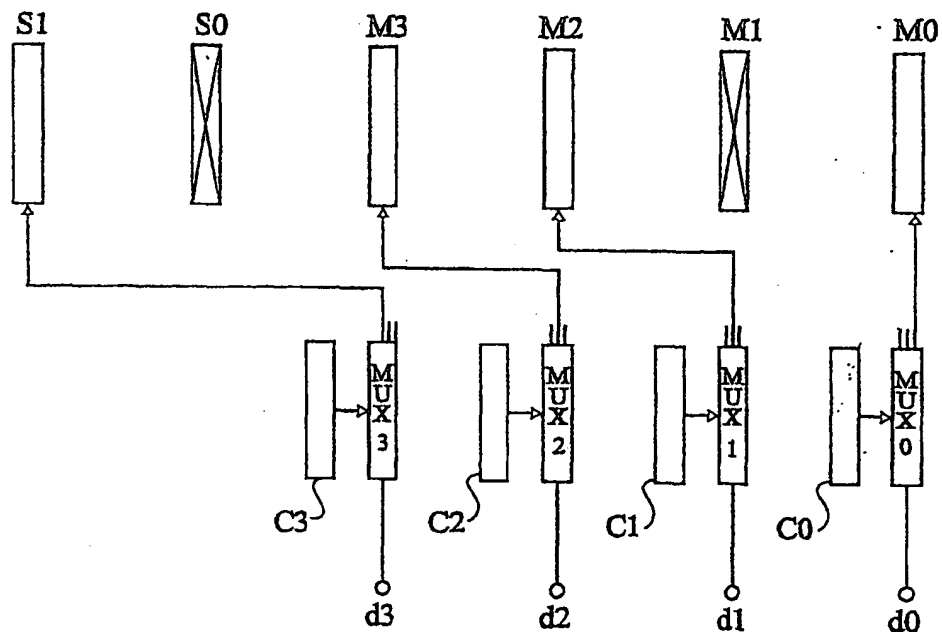


Fig 1A

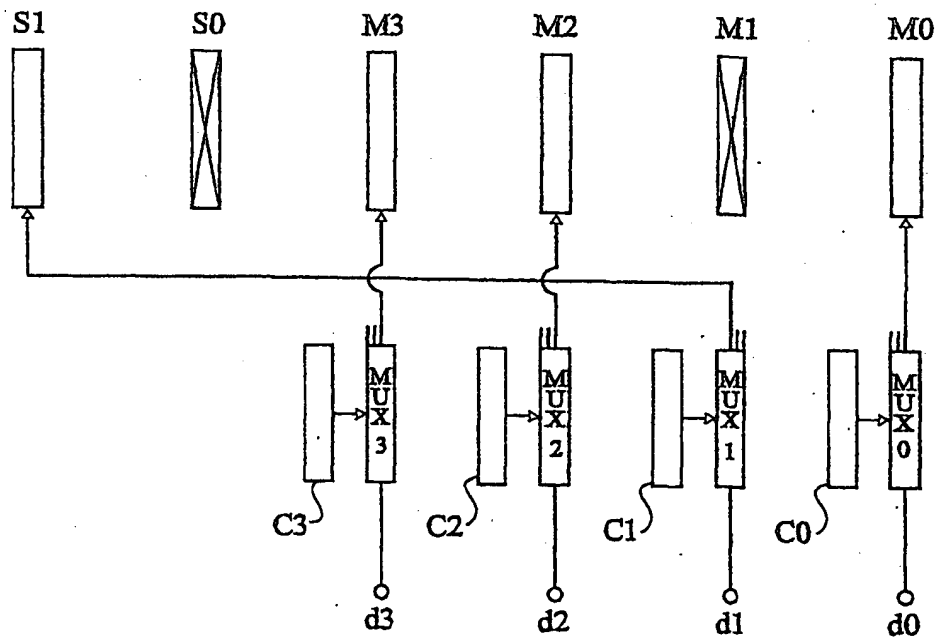


Fig 1B

2/5

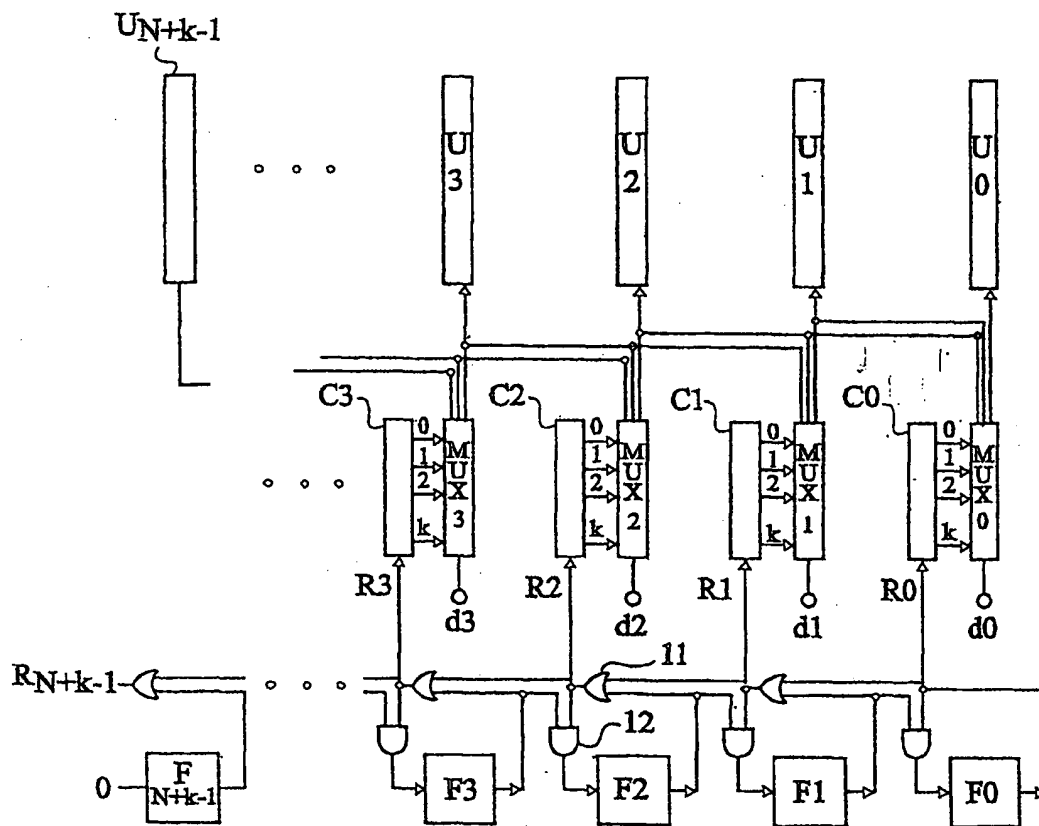


Fig 2

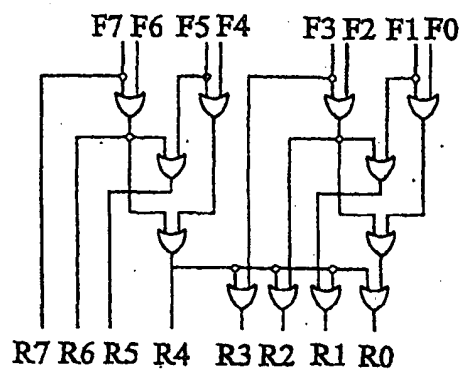
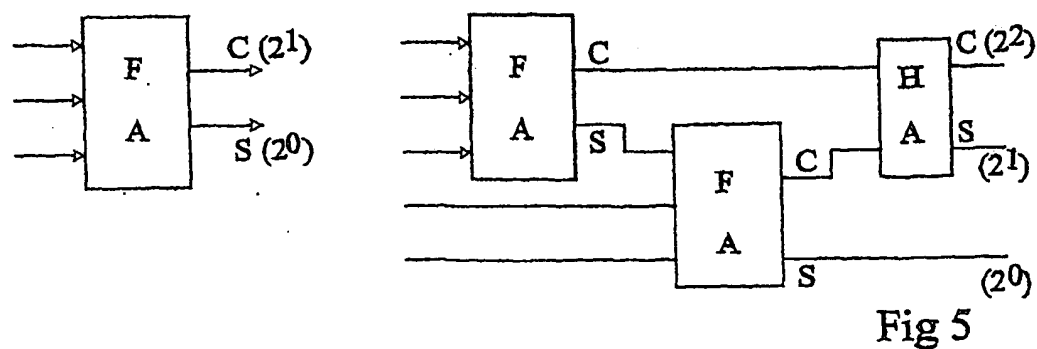
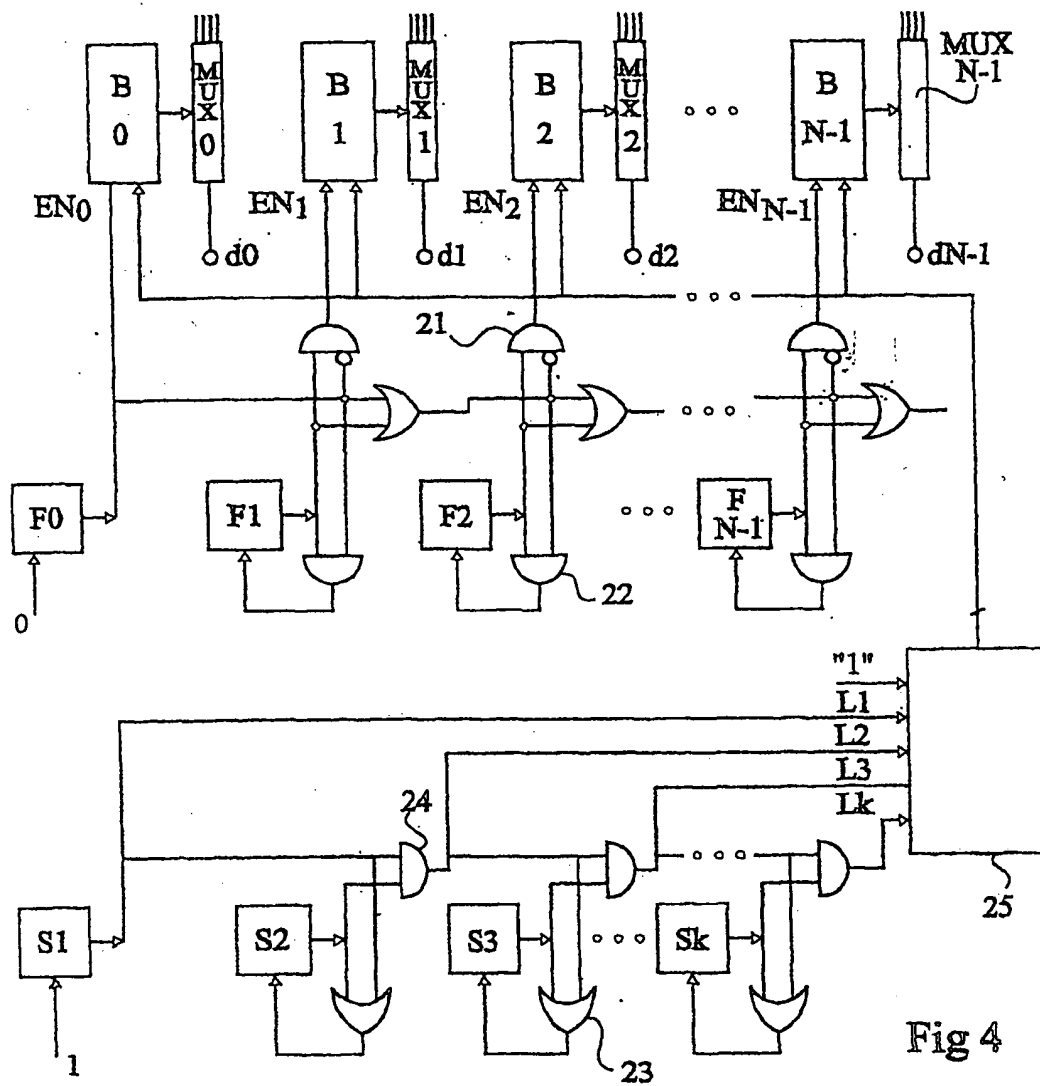


Fig 3

3/5



4/5

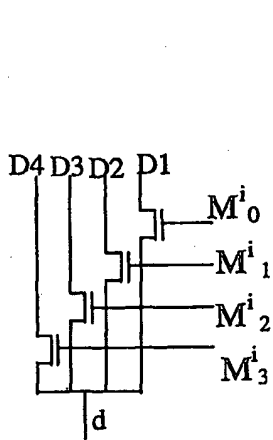


Fig 6

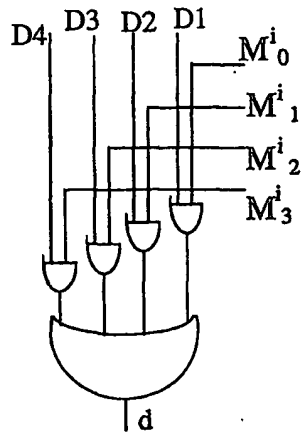


Fig 7

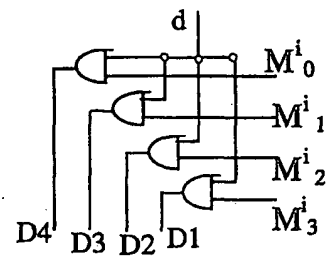


Fig 8

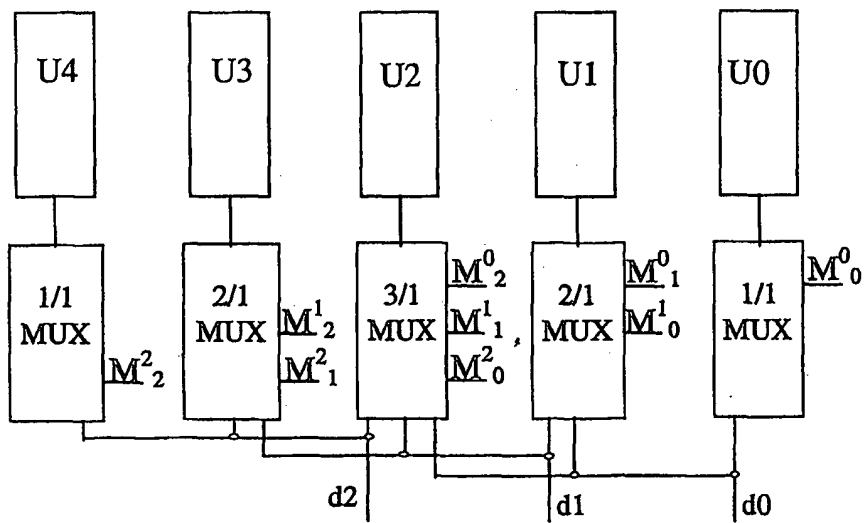


Fig 9

5/5

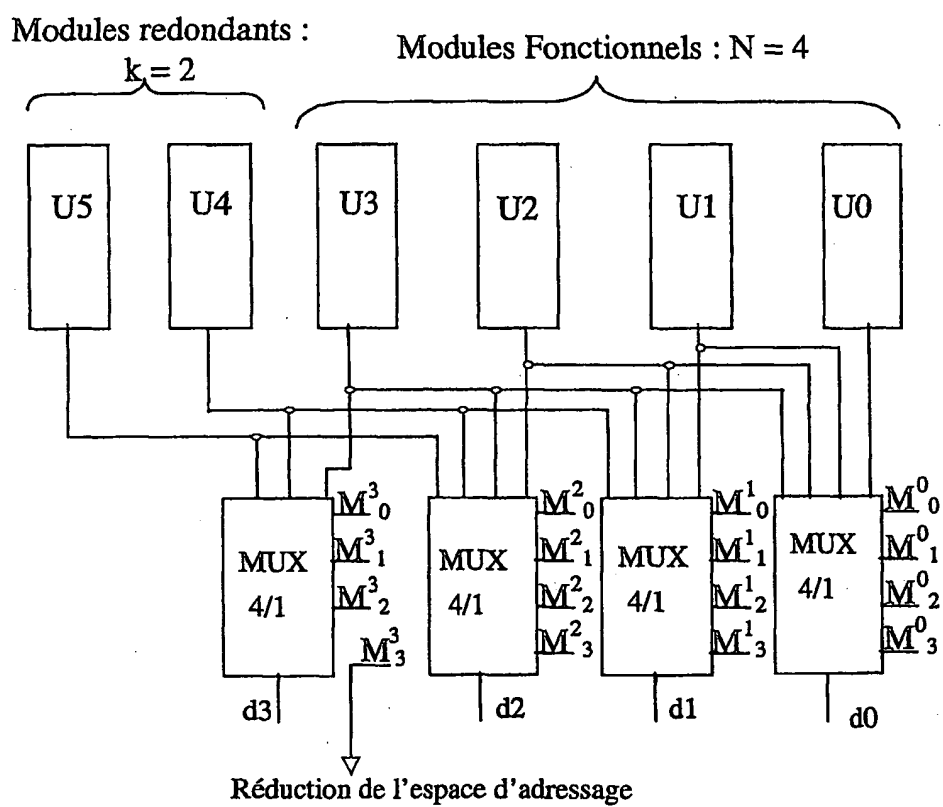


Fig 10

(12) DEMANDE INTERNATIONALE PUBLÉE EN VERTU DU TRAITÉ DE COOPÉRATION
EN MATIÈRE DE BREVETS (PCT)

(19) Organisation Mondiale de la Propriété
Intellectuelle
Bureau international



(43) Date de la publication internationale
22 août 2002 (22.08.2002)

PCT

(10) Numéro de publication internationale
WO 2002/065291 A3

(51) Classification internationale des brevets⁷ :
G06F 11/20, G11C 29/00

(21) Numéro de la demande internationale :
PCT/FR2002/000523

(22) Date de dépôt international :
12 février 2002 (12.02.2002)

(25) Langue de dépôt : français

(26) Langue de publication : français

(30) Données relatives à la priorité :
01/01884 12 février 2001 (12.02.2001) FR

(71) Déposant (pour tous les États désignés sauf US) : IROC
TECHNOLOGIES [FR/FR]; 5, Place Robert Schuman,
F-38000 Grenoble (FR).

(72) Inventeur; et

(75) Inventeur/Déposant (pour US seulement) : NICO-
LAIDIS, Michaël [GR/FR]; 15 bis, rue du Vercors,
F-38120 Saint Egrève (FR).

(74) Mandataire : DE BEAUMONT, Michel; Cabinet Michel
de Beaumont, 1, rue Champollion, F-38000 Grenoble (FR).

(81) États désignés (national) : JP, US.

(84) États désignés (régional) : brevet européen (AT, BE, CH,
CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT,
SE, TR).

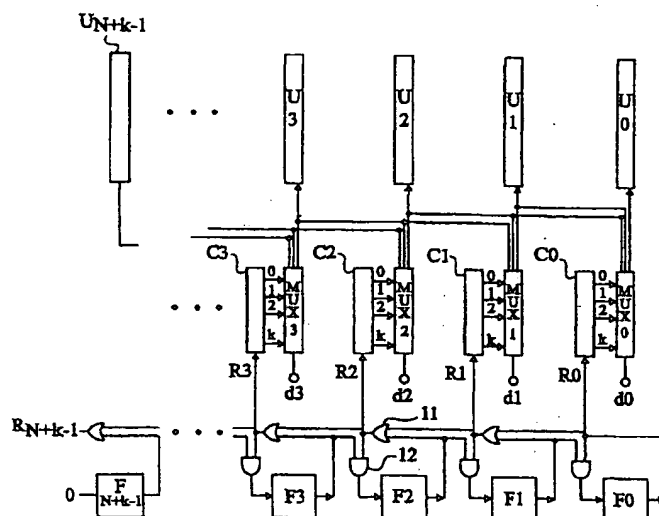
Publiée :

— avec rapport de recherche internationale
— avant l'expiration du délai prévu pour la modification des
revendications, sera republiée si des modifications sont re-
çues

[Suite sur la page suivante]

(54) Title: DEVICE FOR RECONFIGURING A FAULTY STORAGE ASSEMBLY

(54) Titre : DISPOSITIF DE RECONFIGURATION D'UN ENSEMBLE MEMOIRE PRESENTANT DES DEFAUTS



(57) Abstract: The invention concerns a device for reconfiguring an assembly of N basic electronic modules associated with k redundant modules, comprising: N multiplexers each having a first terminal (d_i) capable of being connected to one of the $k+1$ second terminals connected to the $k+1$ input/output terminals of a sequenced group of modules consisting of a basic module (U_i) and k other modules; $N+k$ triggers (F_j) indicating a good or faulty condition of one of the $N+k$ modules; and logic means associated with each multiplexer of rank j, j being an integer ranging between 0 and N, to determine the number of triggers of rank 0 to j indicating a faulty condition, to determine the number of modules of the sequenced group associated with the module of rank j, to be counted to find a number of good modules equal to the first number, and to connect the first terminal of the multiplexer to its second terminal of rank equal to the second number.

[Suite sur la page suivante]

(88) Date de publication du rapport de recherche

internationale:

19 août 2004

En ce qui concerne les codes à deux lettres et autres abréviations, se référer aux "Notes explicatives relatives aux codes et abréviations" figurant au début de chaque numéro ordinaire de la Gazette du PCT.

(57) **Abrége :** L'invention concerne un dispositif de reconfiguration d'un ensemble de N modules électroniques de base associés à k modules redondants, comprenant: N multiplexeurs ayant chacun une première borne (d_i) susceptible d'être connectée à l'une de $k+1$ deuxièmes bornes connectées aux $k+1$ bornes d'entrée/sortie d'un groupe ordonné de modules composé d'un module de base (U_i) et de k autres modules; $N+k$ bascules (F_i) indicatrices d'un état bon ou mauvais d'un des $N+k$ modules; et un moyen logique associé à chaque multiplexeur de rang j , j étant un entier compris entre 0 et N , pour déterminer le nombre de bascules de rang 0 à j indiquant un état mauvais, déterminer le nombre de modules du groupe ordonné associé au module de rang j , à compter pour trouver un nombre de modules bons égal au premier nombre, et connecter la première borne du multiplexeur à sa deuxième borne de rang égal au deuxième nombre.

INTERNATIONAL SEARCH REPORT

International Application No
PCT/FR 02/00523

A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 G06F11/20 G11C29/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 G06F G11C

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the International search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 6 151 263 A (KYUNG KYE-HYUN ET AL) 21 November 2000 (2000-11-21) the whole document	1

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the International filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the International filing date but later than the priority date claimed

- *T* later document published after the International filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- *Z* document member of the same patent family

Date of the actual completion of the International search

29 June 2004

Date of mailing of the International search report

12/07/2004

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax (+31-70) 340-3016

Authorized officer

Absalom, R

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/FR 02/00523

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6151263	A	21-11-2000	
		KR 252053 B1	01-05-2000
		DE 19823584 A1	10-06-1999
		FR 2772507 A1	18-06-1999
		GB 2369909 A ,B	12-06-2002
		GB 2369910 A ,B	12-06-2002
		GB 2332292 A ,B	16-06-1999
		JP 11176187 A	02-07-1999
		TW 402722 B	21-08-2000
		US 6304500 B1	16-10-2001

RAPPORT DE RECHERCHE INTERNATIONALE

Demande Internationale No
PCT/FR 02/00523

A. CLASSEMENT DE L'OBJET DE LA DEMANDE
CIB 7 G06F11/20 G11C29/00

Selon la classification internationale des brevets (CIB) ou à la fois selon la classification nationale et la CIB

B. DOMAINES SUR LESQUELS LA RECHERCHE A PORTE

Documentation minimale consultée (système de classification suivi des symboles de classement)
CIB 7 G06F G11C

Documentation consultée autre que la documentation minimale dans la mesure où ces documents relèvent des domaines sur lesquels a porté la recherche

Base de données électronique consultée au cours de la recherche internationale (nom de la base de données, et si réalisable, termes de recherche utilisés)
EPO-Internal, WPI Data, PAJ

C. DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie *	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
A	US 6 151 263 A (KYUNG KYE-HYUN ET AL) 21 novembre 2000 (2000-11-21) le document en entier	1

☐ Voir la suite du cadre C pour la fin de la liste des documents

☒ Les documents de familles de brevets sont indiqués en annexe

* Catégories spéciales de documents cités:

- *A* document définissant l'état général de la technique, non considéré comme particulièrement pertinent
- *E* document antérieur, mais publié à la date de dépôt international ou après cette date
- *L* document pouvant jeter un doute sur une revendication de priorité ou cité pour déterminer la date de publication d'une autre citation ou pour une raison spéciale (telle qu'indiquée)
- *O* document se référant à une divulgation orale, à un usage, à une exposition ou tous autres moyens
- *P* document publié avant la date de dépôt international, mais postérieurement à la date de priorité revendiquée

- *T* document ultérieur publié après la date de dépôt international ou la date de priorité et n'appartenant pas à l'état de la technique pertinent, mais cité pour comprendre le principe ou la théorie constituant la base de l'invention
- *X* document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme nouvelle ou comme impliquant une activité inventive par rapport au document considéré isolément
- *Y* document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme impliquant une activité inventive lorsque le document est associé à un ou plusieurs autres documents de même nature, cette combinaison étant évidente pour une personne du métier
- *Z* document qui fait partie de la même famille de brevets

Date à laquelle la recherche internationale a été effectivement achevée

29 juin 2004

Date d'expédition du présent rapport de recherche internationale

12/07/2004

Nom et adresse postale de l'administration chargée de la recherche internationale
Office Européen des Brevets, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Fonctionnaire autorisé

Absalom, R

RAPPORT DE RECHERCHE INTERNATIONALE

Renseignements relatifs

membres de familles de brevets

Demande Internationale No

PCT/FR 02/00523

Document brevet cité au rapport de recherche	Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
US 6151263	A	21-11-2000	
		KR 252053 B1	01-05-2000
		DE 19823584 A1	10-06-1999
		FR 2772507 A1	18-06-1999
		GB 2369909 A ,B	12-06-2002
		GB 2369910 A ,B	12-06-2002
		GB 2332292 A ,B	16-06-1999
		JP 11176187 A	02-07-1999
		TW 402722 B	21-08-2000
		US 6304500 B1	16-10-2001